Successive Column Correction Algorithms for Solving Sparse Nonlinear Systems of Equations[1]

by

Guangye Li[2]

Technical Report 86-12, May 1986.

[1] This research was partially supported by contracts and grants: DOE DE-AS05-82ER1-13016 and AFOSR 85-0243. This work was done toward partial fulfillment of the requirements of the Ph. D. degree in Mathematical Sciences Department, Rice University.

[2] Computer Science Department, Cornell University, Upson Hall, Ithaca, New York 14853. Permanent address: Computer Center, Jilin University, People's Republic of China.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **MAY 1986**	2. REPORT TYPE	3. DATES COVERED **00-00-1986 to 00-00-1986**

4. TITLE AND SUBTITLE **Successive Column Correction Algorithms for Solving Sparse Nonlinear Systems of Equations**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Computational and Applied Mathematics Department ,Rice University,6100 Main Street MS 134,Houston,TX,77005-1892**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**24**	

Abstract

This paper presents two algorithms for solving sparse nonlinear systems of equations: the CM-successive column correction algorithm and the modified CM-successive column correction algorithm. A q-superlinear convergence theorem and an r-convergence order estimate are given for both algorithms. The numerical results indicate that these two algorithms, especially the modified algorithm are probably more efficient than some currently used algorithms.

1. Introduction.

Consider a nonlinear system of equations

$$F(x) = 0 \ , \tag{1.1}$$

where F: $R^n \to R^n$ is continuously differentiable on an open convex set $D \subset R^n$, and the Jacobian matrix $F'(x)$ is sparse. To solve the system, the following iteration is considered:

$$x^{k+1} = x^k - B_k^{-1} F(x^k) \ , \qquad k=0,1,..., \tag{1.2}$$

where B_k is an approximation to the Jacobian with the same sparsity structure.

For convenience, we rewrite (1.2) as

$$\bar{x} = x - B^{-1} F(x) \ , \tag{1.3}$$

where x and $\bar{x}$ indicate the current iterate and the new iterate respectively, and B is an approximation to the Jacobian.

Currently, there are several algorithms to get a sparse approximation to the Jacobian. In this paper we will discuss three types of algorithms.

(1) Schubert's algorithm. In 1970 Schubert [17] gave a sparse modification of Broyden's update. Broyden [2] also gave this algorithm independently. In order to present Schubert's algorithm, we introduce the following notation concerning the sparsity pattern of the Jacobian:

Definition 1.1. For $j=1,2,...$, define the subspace $Z_j \subset R^n$ determined by the sparse pattern of the jth row of the Jacobian:

$$Z_j \equiv \{ v \in R^n \colon \ e_i^T v = 0 \text{ for } \textit{all } i \textit{ such that } [F'(x)]_{ji} = 0 \text{ for } \textit{all } x \in R^n \},$$

where e_i is the ith column of the $n \times n$ identity matrix. Define the set of matrices Z that preserve the sparsity pattern of the Jacobian:

$$Z \equiv \{ A \in L(R^n) \colon \ A^T e_j \in Z_j \ \text{ for } j=1,2,...,n \ \}.$$

Definition 1.2. For $j=1,2,...,n$, define the projection operator, $D_j \in L(R^n)$, that maps R^n onto Z_j:

$$D_j \equiv \textit{diag} \ (d_{j1}, d_{j2}, \ . \ . \ . \ , d_{jn}),$$

where

$$\mathrm{d}_{ji} = \begin{cases} 1, & \text{if } e_i \in Z_j, \\ 0, & \textit{otherwise.} \end{cases}$$

For a scalar $\alpha\in R$, define the pseudo-inverse:

$$\alpha = \begin{cases} \alpha^{-1}, & \text{if } \alpha \neq 0, \\ 0, & \text{if } \alpha = 0. \end{cases}$$

Now Schubert's update can be written as

$$\bar{B} = B + \sum_{j=1}^{n}([s]_j^T[s]_j)^+ e_j e_j^T(y - Bs)[s]_j^T, \tag{1.4}$$

where $[s]_j = D_j s$, $s = \bar{x} - x$ and $y = F(\bar{x}) - F(x)$.

Let

$$Q_{u,v} = \{A\in L(R^n)\text{: } Au=v, \text{ for } \textit{vectors } u,\ v\in R^n\}.$$

The following theorem, which we will use later, was proved by Reid [15] and Marwil [9] independently.

Theorem 1.1. Let $B\in Z$; y, $s\in R^n$ with $s\neq 0$. Define $\bar{B}$ by (1.4). Then $\bar{B}$ is the unique solution to

$$\min\{\,||\hat{B} - B\,||_F : \hat{B}\in Q_{y,s}\cap Z\}, \tag{1.5}$$

where $||.||_F$ indicates the Frobenius norm of a matrix.

The advantage of Schubert's algorithm is that at each iteration only one function value is required and it is q-superlinearly convergent (see Marwil [9]). However, it frequently requires more iterations than finite difference algorithms. Moreover, it may not be a good approximation to the Jacobian when the problem is badly nonlinear, especially when the current step is far away from the solution. Therefore, $p_k = -B_k^{-1}F(x^k)$ may not be a descent direction of the functional $f(x) = \frac{1}{2}\,||F(x)||^2$, where $||.||$ denotes the l_2 vector norm. In this case, it may be not good to use a line search with Schubert's algorithm.

(2). Finite difference algorithms. In general, a finite difference algorithm can be formulated as follows: obtain direction vectors $d_1, d_2, \ldots, d_p$ such that B can be determined uniquely by the equations

$$Bd_i = F(x+d_i) - F(x), \qquad i=1,2,\ldots,p.$$

In this paper, we assume that it is not convenient to evaluate the function values element by element, instead we only evaluate the value of $F(x)$ as a single entity. This is reasonable since in practice it is very common that the components of $F(x)$ have expensive common subexpressions. In this case, to reduce the number of function evaluations, Curtis, Powell, and Reid [4] proposed a finite difference algorithm, called the CPR algorithm, which is based on a partition of the columns of the Jacobian. Coleman and Moré [3] associate the partition problem with a graph coloring problem and gave some partitioning algorithms which can make the number of the function evaluations optimal or nearly optimal.

Following Coleman and Moré, we give some definitions concerning a partition of the columns of the Jacobian.

Definition 1.3. A partition of the columns of a matrix B is a division of the columns into groups $c_1, c_2, \ldots, c_p$ such that each column belongs to one and only one group.

Definition 1.4. A partition of the columns of a matrix B is consistent with the direct determination of B if whenever b_{ij} is a nonzero element of B, then the group containing column j has no other column with a nonzero element in row i.

As an example we consider the tridiagonal structure

$$\begin{bmatrix} \times & \times & 0 & 0 & 0 & 0 \\ \times & \times & \times & 0 & 0 & 0 \\ 0 & \times & \times & \times & 0 & 0 \\ 0 & 0 & \times & \times & \times & 0 \\ 0 & 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & 0 & \times & \times \end{bmatrix} . \tag{1.6}$$

A consistent partition of the columns of the matrix is $c_1 = \{1, 4\}$, $c_2 = \{2, 5\}$, and $c_3 = \{3, 6\}$.

The CPR algorithm now can be formulated as follows: for a given consistent partition of the columns of the Jacobian, obtain vectors $d_1, d_2, \ldots, d_p$ such that B is determined uniquely by the equations

$$Bd_i = F(x+d_i) - F(x) \equiv y_i \qquad i=1,2,\ldots,p \ . \tag{1.7}$$

Notice that for the CPR algorithm, the number of function evaluations at each iteration is $p+1$. Since the partition of the columns of the Jacobian plays an important role in the CPR algorithm, we call the CPR algorithm based on Coleman and Moré's algorithms the CPR-CM algorithm.

For the consistent partition given in example (1.6), if we take

$$d_1 = (\ h\ ,0\ ,0\ ,h\ ,0\ ,0\ )^T,$$

$$d_2 = (\ 0\ ,h\ ,0\ ,0\ ,h\ ,0\ )^T,$$

$$d_3 = (\ 0\ ,0\ ,h\ ,0\ ,0\ ,h\ )^T,$$

then B is determined uniquely and the number of function evaluations required at each iteration is 4.

The advantage of the CPR algorithm is that it usually requires fewer iterations than Schubert's algorithm. However, it requires more function values at each iteration than Schubert's algorithm.

(3). The successive column correction algorithms.

Polak [13] gave a successive column correction algorithm for unconstrained minimization. Feng and Li [7] developed a successive column correction algorithm for nonlinear system of equations, which is called the column-update quasi-Newton method. Using this algorithm, columns of B_k are displaced by differences successively and periodically. At each iteration, only two function values are required, but only one column is displaced.

In this paper, we propose two algorithms: the CM-successive column correction algorithm and the modified CM-successive column correction algorithm. The former is based on Coleman and Moré's algorithm and the column-update algorithm. The latter is a combination of the CM-successive column correction algorithm and Schubert's algorithm. Both algorithms require only two function values at each iterative step. Our numerical results show that the CM-successive column correction algorithms, especially the modified one, are probably more efficient than the CPR algorithm and Schubert's algorithm.

The CM-successive column correction algorithm is given in Section 2. A Kantorovich-type analysis for this algorithm is given in Section 3. A q-superlinear convergence result and an r-convergence order estimate of the CM-successive column correction algorithm are given in Section 4. The modified CM-successive column correction algorithm is given in Section 5. Some numerical results are given in Section 6.

In this paper, for a sparse matrix B, we use M to denote the set of pairs of indices (i, j), where b_{ij} is a structurally nonzero element of B, i.e.

$$M = \{(i, j) : \ b_{ij} \neq 0\} .$$

2. The CM-Successive Column Correction Algorithm and its Properties.

Given a consistent partition of the columns of the Jacobian, which divides the set $\{1, 2, ..., n\}$ into p subsets $c_1, c_2, ..., c_p$, let

$$d^k = \sum_{j \in c_{i_k}} h^k e_j, \tag{2.1}$$

where

$$i_k = k \ (\ mod \ p \ ), \qquad k = 1,2,...,$$

and let

$$y^k = F(x^k + d^k) - F(x^k). \tag{2.2}$$

The CM-successive column correction algorithm can be formulated as follows: If $k \leq p$, then for $j \in c_k$, the jth column of B_k is determined uniquely by the equation

$$B_k d^k = y^k, \tag{2.3}$$

and the other columns of B_k are equal to the corresponding columns of B_{k-1}. If $k > p$, the columns of B_k are displaced as described above successively and periodically. In other words, for $j \in c_{i_k}$, the jth column of B_k is determined uniquely by (2.3), and the other columns of B_k are equal to the corresponding columns of B_{k-1}.

For example (1.6), at the first iteration we displace the first group $c_1 = \{1, 4\}$. At the second iteration we displace the second group $c_2 = \{2, 4\}$. At the third iteration we displace the third group $c_3 = \{3, 6\}$, and then we displace the three groups successively and periodically.

The CM-successive column correction algorithm with a global strategy is given below.

Algorithm 2.1. Given a consistent partition of the columns of the Jacobian, which divides the set $\{1, 2, ..., n\}$ into p subsets $c_1, c_2, ..., c_p$ (for convenience, c_i, $i=1,2,...,p$, indicates both the sets of the columns and the sets of the indices of these columns), and given an $x^0 \in R^n$ and a nonsingular matrix B_0, which has the same sparsity as the Jacobian, at the initial step:

(1). Set $l = 0$.

(2). Solve $B_0 s^0 = -F(x^0)$.

(3). Choose x^1 by $x^1 = x^0 + s^0$, or by a global strategy.

At each iteration $k > 0$:

(1). Choose a scalar h^k.

(2). If $l < p$, then set $l = l + 1$, otherwise set $l = 1$.

(3). Set

$$d^k = \sum_{j \in c_l} h^k e_j.$$

(4). If $j \in c_l$ and $(i, j) \in M$, then set

$$b_{ij}^k = \frac{1}{h^k} e_i^T (F(x^k + d^k) - F(x^k)), \tag{2.4}$$

otherwise set

$$b_{ij}^k = b_{ij}^{k-1},$$

where $B_k = [b_{ij}^k]$.

(5). Solve $B_k s^k = -F(x^k)$.

(6). Choose x^{k+1} by $x^{k+1} = x^k + s^k$, or by a global strategy.

(7). Check for convergence.

Let

$$J_k = \int_0^1 F'(x^k + td^k) dt . \tag{2.5}$$

Then

$$J_k d^k = y^k \ . \tag{2.6}$$

Let $J_k = [J^k_{lm}]$. Since J_k has the same sparsity as the Jacobian, by (2.6), we have that if $(l,\ m) \in M$, then

$$J^k_{lm} = \frac{e_l^T y^k}{h^k} \ , \tag{2.7}$$

where $m \in c_{i_k}$. Comparing (2.7) with (2.4), we have

$$B_k e_j = J_k e_j \ ,$$

for $j \in c_{i_k}$.

The CM-successive column correction algorithm is also an update algorithm, and the update can be written as:

$$B_k = B_{k-1}(I - \sum_{j \in c_{i_k}} e_j e_j^T) + \sum_{j \in c_{i_k}} J_k e_j e_j^T. \tag{2.8}$$

From (2.8), it is easy to get the following result:

Lemma 2.2. Let $B_k, \quad k=1,2,...,$ be generated by Algorithm 2.1. If $k \geq p$, then

$$B_k = \sum_{j=k-p+1}^{k} \ \sum_{l \in c_{i_j}} J_j e_l e_l^T. \tag{2.9}$$

To study the properties of our algorithms, sometimes we assume that F' satisfies the following Lipschitz condition: there exist $\alpha_i > 0$, $i=1,2,...,n$ such that

$$||(F'(x) - F'(y))e_i|| \leq \alpha_i ||x - y||, \qquad x,y \in D. \tag{2.10}$$

Let $\alpha = (\sum_{i=1}^{n} \alpha_i^2)^{\frac{1}{2}}$. Then, it follows from (2.10) that

$$||F'(x) - F'(y)||_F \leq \alpha ||x - y||, \qquad x,y \in D. \tag{2.11}$$

Theorem 2.3. Let F' satisfy Lipschitz condition (2.10). Also let $\{x_j\}_{j=0}^{k} \subset D$ and let $\{B_j\}_{j=0}^{k}$ be generated by Algorithm 2.1 with $|h^k| \leq \frac{2}{\sqrt{n}} ||x^k - x^{k-1}||$. If $\{x^j + d^j\}_{j=1}^{k} \subset D$, then for $k \geq p$,

$$||B_k - F'(x_k)||_F \leq \alpha \sum_{j=k-p+1}^{k} ||x^j - x^{j-1}|| \ . \tag{2.12}$$

Proof. By (2.5), (2.1) and Lipschitz condition (2.10),

$$\begin{aligned}
&||(F'(x^m) - J_m)e_j|| \\
&= ||(\int_0^1 (F'(x^m+td^m) - F'(x^m))dt)e_j|| \\
&\leq \alpha_j \int_0^1 ||d^m||\, tdt = \frac{\alpha_j}{2}||d^m|| \qquad (2.13) \\
&= \frac{\alpha_j}{2}||\sum_{j\in c_{i_m}} h^m e_j|| \\
&\leq \frac{\alpha_j}{2}\sqrt{n}\,|h^m| \leq \alpha_j ||x^m - x^{m-1}|| \ ,
\end{aligned}$$

where $k-p+1 \leq m \leq k$. It follows from (2.9) and (2.13) that

$$\begin{aligned}
&||F'(x^k) - B_k||_F^2 \\
&= \sum_{m=k-p+1}^{k} ||\sum_{j\in c_{i_m}} (F'(x^k) - B_k)e_j e_j^T||_F^2 \\
&= \sum_{m=k-p+1}^{k} \sum_{j\in c_{i_m}} ||(F'(x^k) - J_m)e_j||^2 \\
&\leq \sum_{m=k-p+1}^{k} \sum_{j\in c_{i_m}} (||(F'(x^k) - F'(x^m))e_j|| + ||(F'(x^m)-J_m)e_j||)^2 \\
&\leq \sum_{m=k-p+1}^{k} \sum_{j\in c_{i_m}} \alpha_j^2(||x^k - x^m|| + ||x^m - x^{m-1}||)^2 \qquad (2.14) \\
&\leq \sum_{m=k-p+1}^{k} \sum_{j\in c_{i_m}} \alpha_j^2(\sum_{l=k-p+1}^{k} ||x^l - x^{l-1}||)^2 \\
&= \alpha^2(\sum_{l=k-p+1}^{k} ||x^l - x^{l-1}||)^2 \, .
\end{aligned}$$

Then, (2.12) follows from (2.14).

To start iteration (1.2) for a given $x^0 \in D$, an initial matrix B_0 is needed. We suggest using the CPR-CM algorithm to get B_0 since it is easy to implement after we have a consistent partition of the columns of the Jacobian.

3. A Kantorovich-Type Analysis.

By means of Theorem 2.3, we have the following Kantorovich-type analysis for the CM-successive column correction algorithm.

Theorem 3.1. Assume that $F'(x)$ satisfies Lipschitz condition (2.10). Let $x^0 \in D$, and let B_0 be a nonsingular $n \times n$ matrix such that

$$||B_0 - F'(x^0)||_F \leq \delta, \quad ||B_0^{-1}||_F \leq \beta, \quad ||B_0^{-1}F(x^0)|| \leq \eta,$$

$$h = \frac{\alpha\beta\eta}{(1-3\beta\delta)^2} \leq \frac{1}{6}, \tag{3.1}$$

and

$$\beta\delta < \frac{1}{3} .$$

If $\bar{S}(x^0, 2t^*) \subset D$, where

$$t^* = \frac{1-3\beta\delta}{3\alpha\beta}(1-\sqrt{1-6h}) , \tag{3.2}$$

then $\{x^k\}$, generated by the CM-successive column correction algorithm with $|h^k| \leq \frac{2}{\sqrt{n}} ||x^k - x^{k-1}||$ and without any global strategy, converges to x^*, which is the unique root of $F(x)$ in $\bar{S}(x^0, \bar{t}) \cap D$, where

$$\bar{t} = \frac{1-\beta\delta}{\alpha\beta}\left[1 + \left(1-\frac{2\alpha\beta\eta}{(1-\beta\delta)^2}\right)^{\frac{1}{2}}\right] .$$

Proof. Consider the scalar iteration

$$t_{k+1} - t_k = \beta f(t_k), \quad t_0 = 0, \quad k = 0, 1, 2, \cdots , \tag{3.3}$$

where

$$f(t) = \frac{3}{2}\alpha t^2 - (\frac{1-3\beta\delta}{\beta})t + \frac{\eta}{\beta} . \tag{3.4}$$

It is easy to show that the sequence $\{t_k\}$ satisfies the difference equation

$$t_{k+1} - t_k = 3\beta\,[\frac{\alpha}{2}(t_k - t_{k-1}) + \alpha t_{k-1} + \delta]\,(t_k - t_{k-1}) , \quad k=1, 2, \cdots . \tag{3.5}$$

From this equation, it follows that $\{t_k\}$ is a monotonically increasing sequence and

$$\lim_{k\to\infty} t_k = t^* \quad ,$$

where t^* is the smallest root of $f(t)$.

Now, by induction, we will prove that

$$||x^{k+1}-x^k|| \leq t_{k+1}-t_k\,, \quad k=1,2,\cdots\,, \tag{3.6}$$

$$\{x^k\} \subset \bar{S}(x^0,t^*), \quad k=1,2,\cdots\,, \tag{3.7}$$

$$\{x^k+d^k\} \subset \bar{S}(x^0,2t^*)\,, \tag{3.8}$$

and

$$||B_k^{-1}|| \leq 3\beta\,, \quad k=1,2,\cdots\,. \tag{3.9}$$

For $k=0$, we have

$$||x^1-x^0|| \leq \eta = t_1-t_0 \leq t^*\,.$$

Thus,

$$||x^1+d^1-x^0|| \leq ||x^1-x^0|| + ||d^1|| \leq 2||x^1-x^0|| \leq 2t^*\,.$$

Suppose (3.6) holds for $k=0,1,\ldots,m-1$. Then,

$$||x^m-x^0|| \leq \sum_{i=0}^{m-1}(t_{i+1}-t_i) = t_m \leq t^*\,.$$

Therefore, $x^m \in \bar{S}(x^0,t^*)$, and

$$\{x^m+d^m\} \subset \bar{S}(x^0,2t^*)\,.$$

From the proof of Theorem 2.3, it can be seen that for all k,

$$||B_k-F'(x_k)||_F \leq ||B_0-F'(x^0)||_F + \alpha\sum_{j=0}^{k}||x^j-x^{j-1}||\,. \tag{3.10}$$

Therefore,

$$\begin{aligned}
&||B_0^{-1}(B_m-B_0)||\\
&\leq ||B_0^{-1}||_F(||B_m-F'(x^m)||_F + ||F'(x^m)-F'(x^0)||_F + ||F'(x^0)-B_0||_F)\\
&\leq \beta(2\alpha\sum_{i=0}^{m-1}||x^{i+1}-x^i|| + 2\delta)\\
&\leq \beta(2\alpha t^* + 2\delta) \leq \beta(\frac{2/3}{\beta}) = \frac{2}{3}\,.
\end{aligned}$$

Thus, by Dennis and Schnabel's Theorem 3.1.4 [6, p.45],

$$||B_m^{-1}|| \leq \frac{\beta}{1-2/3} = 3\beta \ .$$

Hence,

$$\begin{aligned} & ||x^{m+1}-x^m|| \\ \leq\ & ||B_m^{-1}||_F\, ||F(x^m)-F(x^{m-1})-B_{m-1}(x^m - x^{m-1})|| \\ \leq\ & 3\beta[\frac{\alpha}{2}||x^m - x^{m-1}|| + \alpha \sum_{i=0}^{m-2} ||x^{i+1}-x^i|| + \delta]\, ||x^m - x^{m-1}|| \\ \leq\ & 3\beta[\frac{\alpha}{2}(t_m - t_{m-1}) + \alpha\, t_{m-1} + \delta](t_m - t_{m-1}) = t_{m+1} - t_m \ . \end{aligned}$$

This completes the induction step. By (3.6), it is easy to show that there is an $x^* \in D$ such that

$$\lim_{k\to\infty} x^k = x^* \ .$$

The uniqueness of x^* in $\overline{S}(x^0, \overline{t}\,) \cap D$ can be obtained from Ortega and Rheinboldt's Theorem 12.6.4 [12, p.425] by setting $A(x) \equiv B_0$.

4. Local Convergence Properties.

To study the local convergence of our algorithms, we assume that $F{:}D \subset R^n \to R^n$ has the following property:

$$\textit{There is an } x^* \in D\textit{, such that } F(x^*) = 0 \textit{ and } F'(x^*) \textit{ is nonsingular.} \tag{4.1}$$

Theorem 4.1. Let $F{:}D \subset R^n \to R^n$ satisfy (4.1), and let F' satisfy Lipschitz condition (2.10). Also let $\{x^k\}$ be generated by Algorithm 2.1 with $|h^k| \leq \frac{2}{\sqrt{n}} ||x^k - x^{k-1}||$ and without any global strategy: Then, there exist ϵ, $\delta > 0$ such that if $x^0 \in D$ and B_0 satisfy

$$||x^0 - x^*|| < \epsilon, \quad ||B_0 - F'(x^*)||_F \leq \delta ,$$

then $\{x^k\}$ is well defined and converges q-superlinearly to x^*.

Proof. Notice that when ϵ and δ are small enough, we have that $h \leq \frac{1}{6}$, $\beta\delta < \frac{1}{3}$ and that $\overline{S}(x^0, 2t^*) \subset D$ where h, β and t^* are defined in Theorem 3.1. Therefore, by Theorem 3.1,

$$x^k + d^k \in D \,, \quad k=0,\, 1,\, \cdots \,.$$

By (2.8),

$$\begin{aligned} & B_k - F\,'(x^*) \\ &= (B_{k-1} - F\,'(x^*))(I - \sum_{j \in c_{i_k}} e_j e_j^T) + \sum_{j \in c_{i_k}} (J_k - F\,'(x^*))e_j e_j^T. \end{aligned} \tag{4.2}$$

Thus,

$$\begin{aligned} & \| J_k - F\,'(x^*) \|_F \\ &= \| \int_0^1 (F\,'(x^k - td^k) - F\,'(x^*))dt \|_F \\ &\leq \alpha(\| x^k - x^* \| + \frac{1}{2} \| d^k \|) \\ &\leq \alpha(\| x^k - x^* \| + \| x^k - x^{k-1} \|) \\ &\leq \alpha(2 \| x^k - x^* \| + \| x^{k-1} - x^* \|). \end{aligned} \tag{4.3}$$

Let $\sigma(x^{k-1}\,,\, x^k) = \max\,\{ \| x^k - x^* \| \,,\, \| x^{k-1} - x^* \| \}$. Then it follows from (4.2) and (4.3) that

$$\begin{aligned} & \| B_k - F\,'(x^*) \|_F \leq \| B_{k-1} - F\,'(x^*) \|_F + \| J_k - F\,'(x^*) \|_F \\ &\leq \| B_{k-1} - F\,'(x^*) \|_F + 3\alpha\sigma(x^{k-1}\,,\, x^k)\,. \end{aligned}$$

Thus, by Dennis and More's [5] Theorem 5.1, we know that $\{x^k\}$ converges at least q-linearly to x^*.

According to Dennis and More's [5] Theorem 3.1, to get q-superlinear convergence, we need only to prove that

$$\lim_{k \to \infty} \frac{\| (B_k - F\,'(x^*))(x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} = 0\,. \tag{4.4}$$

From (2.12), it follows that

$$\lim_{k \to \infty} \| B_k - F\,'(x^*) \| = 0. \tag{4.5}$$

This implies (4.4).

Theorem 4.2. Assume that F satisfies the hypotheses in Theorem 4.1. Then the r-convergence order of Algorithm 2.1 is not less than τ, where τ is the unique positive root of

$$t^{p+1} - t^p - 1 = 0\,.$$

Proof. Notice that (4.5) implies that there exist k_0 and $\beta > 0$ such that $||B_k^{-1}|| \leq \beta$ for all $k \geq k_0$. Thus, by Theorem 2.3,

$$\begin{aligned}
&||x^{k+1} - x^*|| = ||x^k - x^* - B_k^{-1}F(x^k)|| \\
&\leq ||B_k^{-1}||_F\{||F(x^k) - F(x^*) - F'(x^*)(x^k - x^*)|| \\
&\qquad + (||F'(x^*) - F'(x^k)||_F + ||F'(x^k) - B_k||_F)\,||x^k - x^*||\} \\
&\leq \beta\{\frac{3}{2}\alpha\,||x^k - x^*|| + \alpha \sum_{j=k-p}^{k-1} ||x^{j+1} - x^j||\}\,||x^k - x^*|| \\
&\leq \frac{5}{2}\alpha\beta(\sum_{j=k-p}^{k} ||x^j - x^*||)\,||x^k - x^*|| \quad .
\end{aligned}$$

Thus, the desired result follows from Ortega and Rheinboldt's Theorem 9.2.9 [12, p.291].

5. The Modified CM-Successive Column Correction Algorithm.

Estimate (2.12) shows that when p is small, B_k is a good approximation to $F'(x^k)$. However, B_k still retains information from the previous p steps. Therefore, the following question is reasonable: Can we have a better approximation to $F'(x^k)$ without more function evaluations? Notice that when we get B_k by Algorithm 2.1, we did not use the value of $F(x^k)$. The main idea of the modified CM-successive column correction algorithm stated below is to use all the information we already have to improve our approximation to $F'(x^k)$.

Algorithm 5.1. Given a consistent partition of the columns of the Jacobian, a vector x^0 and a nonsingular matrix B_0 with the same sparsity as the Jacobian, at the initial step:

(1). Set $l = 0$ and $\overline{B}_0 = B_0$.

(2). Solve $\overline{B}_0 s^0 = -F(x^0)$.

(3). Choose x^1 by $x^1 = x^0 + s^0$, or by a global strategy.

At each iteration $k > 0$:

(1). Update B_{k-1} by Algorithm 2.1 to get B_k.

(2). Update B_k by Schubert's update to get $\overline{B}_k$.

(3). Solve $\overline{B}_k s^k = -F(x^k)$.

(4). Choose x^{k+1} by $x^{k+1} = x^k + s^k$, or by a global strategy.

(5). Check for convergence.

Our numerical results show that Algorithm 5.1 usually requires fewer iterations than Algorithm 2.1. Especially, when the problem is not well behaved, and a global strategy is used, the modified algorithm behaves significantly better than Algorithm 2.1. The cost of the improvement is the computation of Schubert's update. However, since the Jacobian is sparse, Schubert's update requires only $O(n)$ operations. We feel that it is worth doing this rather than computing more function values and solving more linear systems.

Now we will briefly discuss the convergence properties of Algorithm 5.1. Let

$$\bar{J}_k = \int_0^1 F'(x^{k-1} + t(x^k - x^{k-1}))dt. \tag{5.1}$$

Since $\bar{J}_k$ performs exactly the same as the secant factor $\frac{f(x^k)-f(x^{k-1})}{x^k-x^{k-1}}$ in one dimensional problems, we call $\bar{J}_k$ the secant operator. It is easy to show the following result.

Lemma 5.1. If F' satisfies Lipschitz condition (2.10), then

$$||\bar{J}_k - F'(x^k)||_F \leq \frac{\alpha}{2} ||x^k - x^{k-1}|| \ . \tag{5.2}$$

Estimate (5.2) shows that $\bar{J}_k$ is a good approximation to $F'(x^k)$ when $||x^k - x^{k-1}||$ is small.

Theorem 5.2. Let F' satisfy Lipschitz condition (2.10). If $\{B_k\}$ and $\{\bar{B}_k\}$ are generated by Algorithm 5.1, then

$$||\bar{B}_k - \bar{J}_k||_F \leq ||B_k - \bar{J}_k||_F. \tag{5.3}$$

If, in addition, $\bar{B}_k \neq B_k$, then the strict inequality holds.

Proof. Since $\bar{J}_k \in Q_{y,s} \cap Z$, by Theorem 1.1, we have

$$||\bar{B}_k - \bar{J}_k||_F^2 + ||\bar{B}_k - B_k||_F^2 = ||B_k - \bar{J}_k||_F^2. \tag{5.4}$$

Then, (5.3) follows from (5.4).

Notice that in general, $\bar{B}_k \neq B_k$. Therefore, by Theorem 5.2, $\bar{B}_k$ is usually closer to the secant operator $\bar{J}_k$ than B_k. Thus, $\bar{B}_k$ should be a better approximation to the Jacobian than B_k

when B_k retains some information from previous steps. But theoretically, we can not get a better estimate for $||\bar{B}_k - F'(x^k)||_F$ than that for $||B_k - F'(x^k)||_F$. However, we can get the following result:

Theorem 5.3. Let $F : R^n \to R^n$ satisfy Lipschitz condition (2.10). Also let $\{B_k\}$ and $\{x^k\}$ be generated by Algorithm 5.1. Then,

$$||\bar{B}_k - F'(x_k)||_F \le 2\alpha \sum_{j=k-p+1}^{k} ||x^j - x^{j-1}|| \ . \tag{5.5}$$

Proof. By (5.3),

$$\begin{aligned} &||\bar{B}_k - F'(x_k)||_F \\ &\le ||\bar{B}_k - \bar{J}_k||_F + ||\bar{J}_k - F'(x^k)||_F \\ &\le ||B_k - \bar{J}_k||_F + ||\bar{J}_k - F'(x^k)||_F \\ &\le ||B_k - F'(x^k)||_F + 2\,||\bar{J}_k - F'(x^k)||_F \ . \end{aligned}$$

Then, from (2.12) and (5.2), we obtain (5.5).

From estimate (5.5), it is easy to prove that Algorithm 5.1 has at least the same local convergence properties as Algorithm 2.1.

6. Numerical Results.

We computed some examples with tridiagonal Jacobians by the CPR algorithm, Schubert's algorithm, Algorithm 2.1, and Algorithm 5.1. In this section, we compare the numerical results from these four algorithms. The global strategy we used in computing the examples is the line search with backtracking strategy (see Dennis and Schnabel [6, p.126]). For the CPR algorithm, if $p^k = -B_k^{-1}F(x^k)$ is not a descent direction, then we try $-p^k$. If it is not a descent direction either, then the algorithm fails. For the other algorithms, if p^k is not a descent direction, then we try $-p_k$. If it is not a descent direction either, then we try the CPR direction. If the CPR direction fails, then the algorithm fails. In the CPR algorithm, Algorithm 2.1 and Algorithm 5.1, at step k, we use different h_j^k for each component of x^k instead of one uniform h^k. According to Dennis and Schnabel [6, p.98], we choose

$$h_j^k = \sqrt{macheps}\ x_j^k.$$

The stopping test we used is

$$\max_{1 \le i \le n} \frac{|x_i^{k+1} - x_i^k|}{\max\{|x_i^{k+1}|,\ typx_i\}} \le \epsilon\ ,$$

and we choose $\epsilon = 10^{-5}$. We used double precision, and the machine precision is $2.22d{-}16$.

Example 6.1 was given by Guangye Li [8], and it can be seen to be an extension of the Rosenbrock [16] function (also see Moré, Garbow and Hillstrom [11]) to nonlinear system of equations with tridiagonal structure. Example 6.2 was given by Broyden [1] (also see Moré, Garbow and Hillstrom [11]). Example 6.3 was given by Moré and Cosnard [10] (also see Moré, Garbow and Hillstrom [11]). The results are shown in the tables below, where IT is the number of iterations, NF is the number of function($F(x)$) evaluations, and LN is the number of line searches in which the step length $\lambda < 1$. ND is the number of nondecrease directions. $x0$ is the initial guess.

Example 6.1.

$f_1(x) = 8(x_1 - x_2^2)$,

$f_j(x) = 16x_j(x_j^2 - x_{j-1}) - 2(1 - x_j) + 8(x_j - x_{j+1}^2)$, $\quad j = 2, \ldots, n-1$,

$f_n(x) = 16x_n(x_n^2 - x_{n-1}) - 2(1 - x_n)$,

$n = 9$,

$x1 = (-1, -1, \ldots, -1)^T$, $x2 = (-0.5, -0.5, \ldots, -0.5)^T$, $x3 = (2, 2, \ldots, 2)^T$.

Algorithms	$x0 = x1$				$x0 = x2$				$x0 = x3$			
	IT	NF	LN	ND	IT	NF	LN	ND	IT	NF	LN	ND
CPR	22	88	15	0	22	88	15	0	8	32	0	0
Schubert	38	41	21	7	53	56	47	5	33	36	13	5
Alg. 2.1	fail				56	114	46	14	13	28	0	0
Alg. 5.1	24	50	14	0	24	50	15	0	14	30	1	0

Table 6.1.

Example 6.2 (Broyden tridiagonal function).

$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1,$

$x_0 = x_{n+1} = 0,$

$n = 9,$

$x1 = (-1, -1, ..., -1)^T, \;\; x2 = (-0.3, 0.3, ..., -0.3, 0.3)^T,$

$x3 = (-10, -10, ..., -10)^T.$

Algorithms	$x0{=}x1$				$x0{=}x2$				$x0{=}x3$			
	IT	NF	LN	ND	IT	NF	LN	ND	IT	NF	LN	ND
CPR	5	20	0	0	6	24	1	0	8	32	0	0
Schubert	7	10	0	0	11	14	2	0	27	30	3	2
Alg. 2.1	6	14	0	0	8	18	2	0	12	26	0	0
Alg. 5.1	6	14	0	0	7	16	2	0	11	24	0	0

Table 6.2.

Example 6.3 (Discrete boundary value function).

$$f_i(x) = 2x_i - x_{i-1} - x_{i+1} + \frac{h^2}{2}(x_i + t_i + 1)^3$$

$$h = \frac{1}{n+1}, \qquad t_i = ih, \qquad x_0 = x_{n+1} = 0.$$

$n = 9,$

$x1 = (\eta_j)^T \;, \;\eta_j = t_j(t_j - 1), \;\; x2 = (-1, -1, ..., -1)^T \;,$

$x3 = (10, 10, ..., 10)^T.$

Algorithms	$x0{=}x1$				$x0{=}x2$				$x0{=}x3$			
	IT	NF	LN	ND	IT	NF	LN	ND	IT	NF	LN	ND
CPR	3	12	0	0	4	16	0	0	8	32	0	0
Schubert	4	7	0	0	5	8	0	0	17	20	2	2
Alg. 2.1	4	10	0	0	6	14	0	0	12	26	0	0
Alg. 5.1	4	10	0	0	5	12	0	0	10	22	0	0

Table 6.3.

7. Concluding Remarks.

We have presented two algorithms for solving sparse nonlinear systems of equations. The CM-successive column correction algorithm (Algorithm 2.1) is based on Coleman and Moré's partitioning algorithm and the column-update algorithm. This algorithm uses only two function values at each iterative step, and it is q-superlinearly convergent. Using this algorithm, one group of the columns of B_k is displaced at each step. Actually, it is not necessary to update just one group at each iterative step. Instead, we can displace several groups at each iteration, and this gives the algorithm a faster convergence rate. However, if one more group is displaced, then one more function value is required. Therefore, the efficiency of the algorithm depends on the number of the groups displaced at each iterative step.

The modified CM-successive column correction algorithm (Algorithm 5.1) is a combination of the CM-successive column correction algorithm and Schubert's algorithm. It is also q-superlinearly convergent. Our numerical results indicate that the modified successive column correction algorithm usually behaves much better than the CM-successive column correction algorithm. However, we have not been able to prove better theoretical convergence results for the modified CM-successive column correction algorithm than those for the unmodified one. Additional numerical results indicate that the modified CM-successive column correction algorithm is also usually more efficient than the CPR-CM algorithm and Schubert's algorithm. When the problem is not well behaved, or the initial guess is far away from the solution, the modified CM-successive column correction algorithm is much more efficient than Schubert's algorithm.

The idea of the CM-successive column correction algorithms can also be used with Powell and Toint's [14] work, which will lead to methods for unconstrained optimization problems. This will be our future work.

Acknowledgement. This paper is part of my Ph. D. thesis. I would like to express my deepest thanks to my advisor John Dennis. He read this paper several times and made many suggestions and corrections which improved this paper a lot. I would also like to thank Jorge Moré and Richard Tapia for their helpful suggestions and corrections.

References

[1]. Broyden, C.G., A class of methods for solving nonlinear simultaneous equations, Math. Comp., 19 (1965), pp. 577-593.

[2]. Broyden, C.G., The convergence of an algorithm for solving sparse nonlinear systems, Math. Comp., 25 (1971), pp. 285-294.

[3]. Coleman, T.F., and J.J. Moré, Estimation of sparse Jacobian and graph coloring problems, SIAM J. Numer. Anal., 20 (1983), pp. 187-209.

[4]. Curtis, A.R., M.J.D. Powell and J.K. Reid, On the estimation of sparse Jacobian matrices, IMA J. Appl. Math., 13 (1973), pp. 117-119.

[5]. Dennis, J.E., Jr., and J.J. Moré, Quasi-Newton Methods, Motivation and Theory, SIAM Review, Vol. 19, No. 1 (1977).

[6]. Dennis, J.E., Jr., and R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, EngleCwood Cliffs, New Jersey (1983).

[7]. Feng, Guocheng, and Guangye Li, Column-update quasi-Newton method, Comp. Math. of Universities, Vol. 5, No. 2 (1983), pp. 139-147, China.

[8]. Li, Guangye, A new algorithm for solving sparse nonlinear systems of equations, Technical Report 86-1, Math Sciences Dept., Rice Univ. (1986).

[9]. Marwil, E., Convergence results for Schubert's method for solving sparse nonlinear equations, SIAM J. Numerical Analysis, Vol. 16, No. 4, (1979).

[10]. Moré, J.J., and M.Y. Cosnard, Numerical solution of nonlinear equations, ACM Tans. Math. Sofw. 5 (1979), pp. 64-85

[11]. Moré, J.J., B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, ACM Transactions on Mathematical Software, Vol. 7, No. 1 (1981), pp. 17-41.

[12]. Ortega, J.M., and W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York (1970).

[13]. Polak, E., A modified secant method for unconstrained minimization, Memorandum No. ERL-M373 (1973), Electronics Research Lab, College of Engineering, University of California, Berkeley.

[14]. Powell M.J.D., and PH.L. Toint, On the estimation of sparse Hessian matrices, SIAM J. Numer. Anal., Vol. 16 (1979), pp. 1060-1074.

[15]. Reid, J.K., Least squares solution of sparse systems of non-linear equations by a modified Marquardt algorithm, Proceedings of the NATO Conf. at Cambridge, July 1972, North Holland, Amsterdam, pp. 437-445.

[16]. Rosenbrock, H.H., An automatic method for finding the greatest or least value of a function, Comput. J. 5 (1962), pp.147-151.

[17]. Schubert, L.K., Modification of a quasi-Newton method for nonlinear equations with a sparse Jacobian, Math. Comp., 24 (1970), pp. 27-30.